Springer

Journal of
Mechanical
Science and
Technology

Journal of Mechanical Science and Technology 23 (2009) 1939~1948

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-009-0509-y

Automatic simplification of solid models for engineering analysis independent of modeling sequences †

Yoonhwan Woo*

Department of Mechanical Systems Engineering, Hansung University, Seoul, 136-792, Korea

(Manuscript Received December 24, 2008; Revised April 27, 2009; Accepted May 4, 2009)

Abstract

Although solid models can represent complex and detailed geometry of parts, it is often necessary to simplify solid models by removing the detailed geometry in some applications such as finite element analysis and similarity assessment of CAD models. There are no standards for judging the goodness of a simplification method, but one essential criterion would be that it should generate a consistent and acceptable simplification for the same solid model, regardless of how the solid model has been created. Since a design-feature-based approach is tightly dependent on modeling sequences and designer's modeling preferences, it sometimes produces inconsistent and unacceptable simplifications. In this paper, a new method is proposed to simplify solid models of machined parts. Independently of user-specified design features, this method directly recognizes and generates subtractive features from the final model of the part, and then simplifies the solid model by removing the detailed geometry by using these subtractive features.

Keywords: History independency; Level of detail; Solid model simplification; Volume decomposition

1. Introduction

With advances in 3D solid modeling technology, the complex and detailed geometry of part or product can be represented by 3D CAD systems. The solid model has a set of rich information that is utilized by many useful applications in product development cycle. However, in some applications the complex and detailed geometry of the model is regarded as a barrier. For example, finite element analysis (FEA) is becoming an integral part of a CAD system. It uses the solid model to generate meshes from the readily available data in the form of a solid model. However, solid models generated by CAD systems are often unsuitable for analysis needs, requiring appropriate detail removal and dimensional reduction [1]. Fig 1. shows an example of simplifications of a solid model.

The need for solid model simplification arises most often in the generation of meshes, where small and detailed features or geometry end up with improper meshes and cause inefficient computational cost. As shown in Fig. 2, meshing the small features may end up with geometric or topological inaccuracies such as

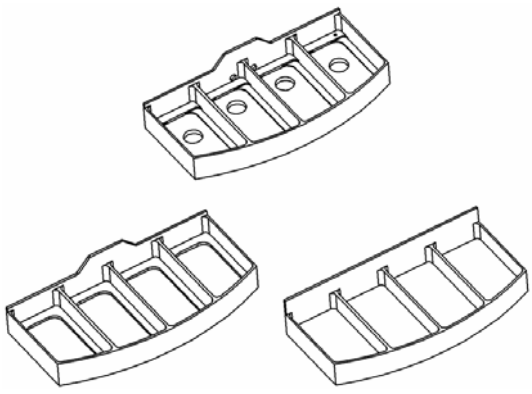


Fig. 1. Different levels of simplifications of solid model of a machined part.

† This paper was recommended for publication in revised form by Associate Editor Jeong-Sam Han

*Corresponding author. Tel.: +82 2 760 4149, Fax.: +82 2 760 4329
E-mail address: yhwoo@hansung.ac.kr

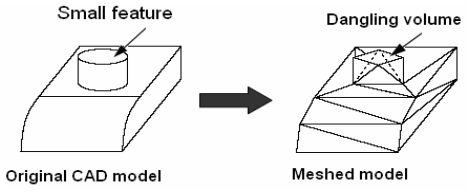


Fig. 2. Undesirable result due to a small feature in mesh generation [2].

self-intersection, gap, and non-manifold edges [2]. Therefore, as far as functional characteristic is maintained, a certain level of idealization needs to be performed and it is mainly achieved by solid model simplification. [1-4]

There have been research efforts to simplify solid models of parts [5-11]. Earlier attempts to remove detailed features focused on recognition and removal of small features from the final models by geometric reasoning. Venkataraman et al. [5] proposed a method to recognize and remove blending features such as fillets. Koo et al. [6] developed the methods called wrap-around and smooth-out. The wrap-around method mimicked the way people wrap things with plastic wraps. To implement this, they employed convex hull generation. But as known, the convex hull generation for a curved object is not easy and has many limitations. The smooth-out algorithm detects small features such as protrusions and depressions that have a closed loop of concave edges. It identifies such features by searching for each feature's unique pattern represented by face-edge graphs. When the small features are identified, it removes the faces of features from the model and fills the gap by extending the neighboring faces. The smooth-out algorithm, however, has a limitation in that it cannot recognize a feature whose concave loop lies on multiple faces as well as intersecting features.

As an attempt to use feature-based CAD systems, Lee J. et al. [9] and Lee S.H. [10] proposed to use design features in the history tree for solid model simplification. The underlying idea of these methods is to reorder the design features in the history tree and then re-execute the history of the reordered features up to a given level of simplification. Since re-evaluation of boundaries of a solid model is computationally heavy, they used a cellular model for performance increase. The cellular model is a non-manifold representation of the feature model geometry, integrating the contributions from all the features used by designer.

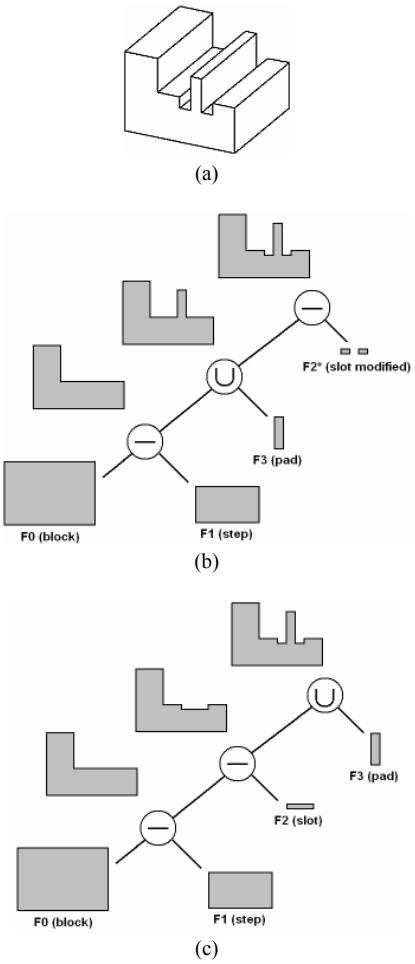


Fig. 3. Simplification by reordering design features; (a) a solid model; (b) original tree of features; (c) reordered tree of features by size.

In reordering design features, Lee, J. et al. [9] start from a base model that is the sum of all the additive design features, order subtractive design features by their volume size, and then sequentially apply the ordered subtractive feature to the base model for solid model simplifications. This approach, however, has a limitation in that a solid model is not processed at all for simplification when the solid model is created only with additive features [10]. Moreover, it has a shortcoming that an additive feature always precedes over a subtractive feature regardless of its volume size.

Compared with Lee J.'s method, Lee S.H. [10] proposed a method to order design features independent of type of feature. However, since the commutative law does not stand in mixed Boolean operations of union and subtraction, re-execution of history of reor-

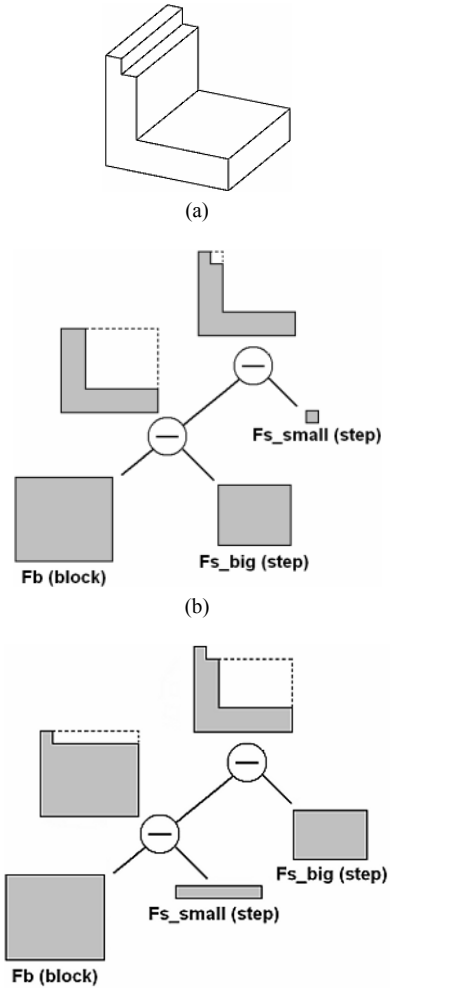


Fig. 4. Identical CAD model with different sets of design features.

dered features may result in a different solid model. To address this problem, he developed a feature algebra that enables the commutative law using the cellular model. Fig. 3 shows an example of simplification by this method.

But these design-feature-based approaches have a critical problem because simplification for a solid model may vary according to designers' modeling sequences and habits. It usually happens when a feature is attached to another feature but they do not intersect. In such case, design-feature-based methods may produce inconsistent simplification for the same model, and sometimes the result is practically unacceptable. This problem of inconsistency is further discussed in the following section.

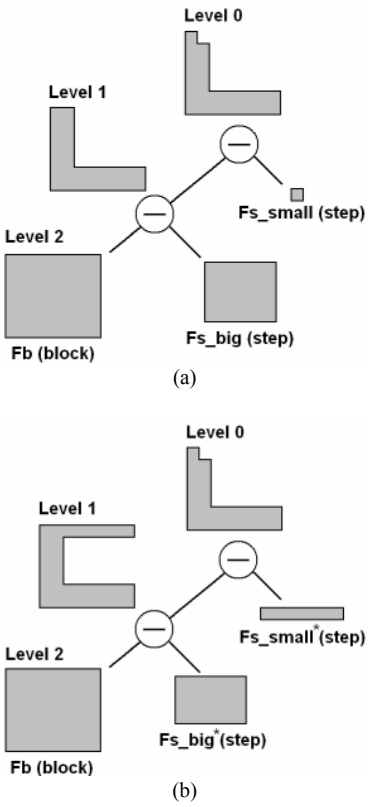


Fig. 5. Simplification by reordering design features in Fig. 4; (a) simplification for the model in Fig. 4(b); (b) simplification for the model in Fig. 4(c).

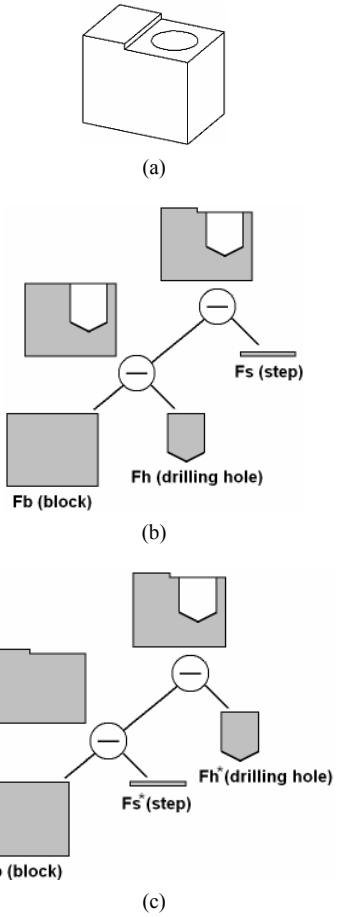
2. Simplification independent of modeling sequence

There are many different ways to create a solid model by using design features. One may create a solid model with a set of design features, and another may create exactly the same model with a different set of design features. Which set of design features to be used relies on a designer's preference and modeling practices.

The solid model in Fig. 4 consists of three features, a block feature and two step features. In Fig. 4(b), the model in Fig. 4(a) is created by using the features in the order of Fb → Fs_big → Fs_small. The same model can also be created by using the features in the order of Fb → Fs_small* → Fs_big* as shown in Fig. 4(c). The sizes of features and the order of features are different in these two cases, but the final geometry of the models is the same. With a design-feature-based simplification method, the order of these features should be rearranged by their size. Ap-

plying this rearrangement to each set of the design features in Fig. 4(b) and Fig. 4(c), we have the reordered histories of the design features as shown in Fig. 5(a) and Fig. 5(b), respectively. Here the results of simplifications obtained by re-executing the reordered histories for these two cases are not identical. Since the design-feature-based method is tightly dependent on user-specified design features, it is not possible to deduce the identical feature interpretations from these different sets of design features.

Another example of this problem is illustrated in Fig. 6. The solid model has three features, a block, a drilling hole, and a step. This model is created by using either the features in the order of Fb → Fh → Fs in Fig. 6(b) or the features in the order of Fb → Fs* → Fh* in Fig. 6(c). The reordered histories for these two cases are shown in Fig. 7(a) and Fig. 7(b), respectively. It is noted that not only are the simplifications not identical, but also unacceptable is the simplified model at level 1 depicted by a circle in Fig. 7(b).



(a)

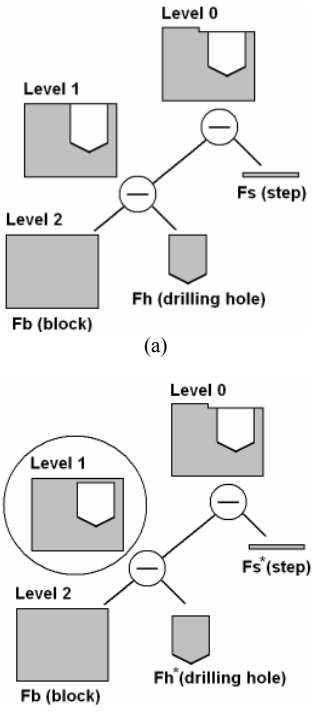
(b)

(c)

Fig. 6. Identical CAD model with different sets of design features.

These inconsistent and sometimes unacceptable results of simplifications occur when the design features do not volumetrically intersect. More precisely, it usually happens when a feature is in contact with another but they do not intersect. Therefore, similar to feature recognition for machining, it is required that a method for solid model simplification has the capability of generation of multiple feature interpretations so that it can produce consistent and more decent simplification results.

Since the design-feature-based approach solely uses the design features specified by designers, it lacks the capability of multiple feature interpretations. This design-feature-based approach may also suffer from the problem of persistent naming [11, 12]. Re-execution of reordered features may end up with an invalid model. In addition, it becomes inapplicable when a feature-based CAD model is translated into a static CAD model such as STEP and IGES. From these observations, it is concluded that the final geometry of the solid model should be used in order to



(a)

(b)

Fig. 7. Simplification by reordering design features in Fig. 6; (a) simplification for the model in Fig. 6(b); (b) simplification for the model in Fig. 6(c).

obtain consistent and acceptable simplification results independent of not only modeling sequences but also the persistent naming problem.

3. Automatic simplification for solid model by decomposition

To accomplish consistent simplification results independent of modeling history, it is necessary to recognize features to be removed directly from the final model. The solid model of a mechanical part usually contains additive and subtractive features. However, when a mechanical part is to be created by machining, it needs to be represented by a set of subtractive features only, each of which corresponds to an appropriate material removal process with a machine tool.

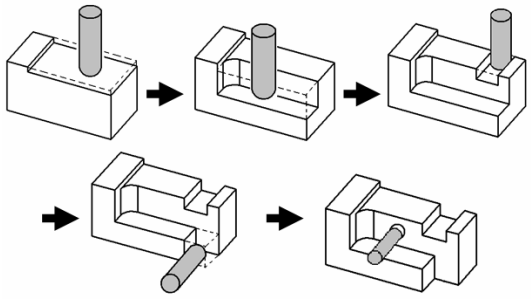


Fig. 8. Example of part creation by machining operations.

Once these subtractive features are recognized, it is easy to use them for simplification for the solid model. For example, a blank workpiece turns into a more finished part as machining operations proceed in Fig. 8.

Similar to this part creation by machining, we propose a method to simplify a solid model by recognizing subtractive features for simplification from the final geometry of the solid model. By doing this, we can obtain consistent results of simplification regardless of user-specified design features. The overview of our method for simplification for the solid model of a machined part is illustrated in Fig. 9. In the following sections, each step of the method is explained in more detail.

3.1 Decomposition of delta volume

There exist many methods to recognize subtractive features from the solid model of a machined part. Feature recognition by graph-matching is relatively fast, but it often fails to recognize intersecting features and additional operation is required to remove faces of recognized features. On the other hand, feature recognition by volume decomposition can recognize intersecting features in the form of solid, although it is relatively slower than the graph-matching method. Since solids of features are readily available upon the completion of feature recognition, feature recognition

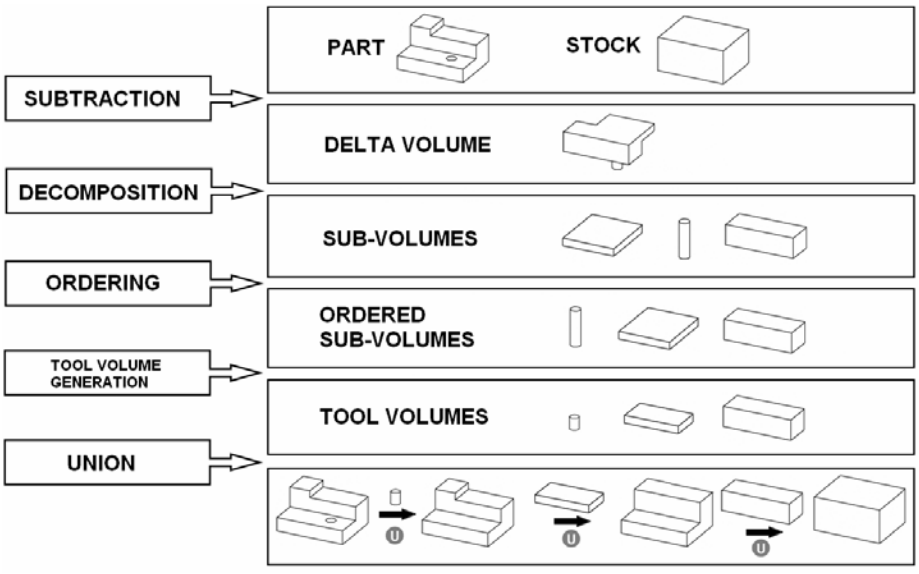


Fig. 9. Overview of simplification for solid model of machined part.

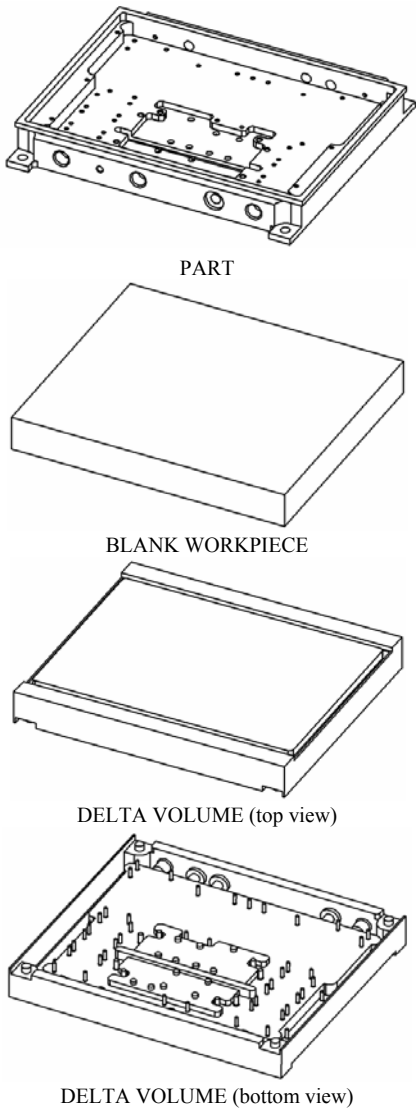


Fig. 10. The blank workpiece and the delta volume of a machined part.

by volume decomposition conforms to the purpose of this research.

In this work, it is assumed that the model at the highest level of simplification is a blank workpiece. A blank workpiece is given by the user, otherwise the bounding box of the part is automatically created and used for a blank workpiece. In creating a part from a blank workpiece, the difference between the part and blank workpiece can be considered as the sum of subtractive features each of which is to be removed by a machining operation. This difference between the part and the blank workpiece is called delta volume and its solid model is simply generated by a Boolean subtraction. Application of maximal volume decomposition to a delta volume decomposes the delta volume into simple sub-volumes that have no concave edges. The maximal volume decomposition is presented in more detail in [13-15].

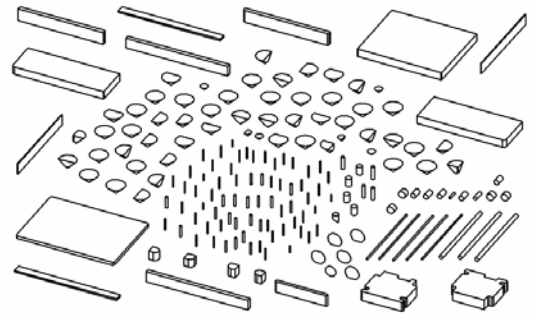


Fig. 11. Sub-volumes of the delta volume in Fig. 10 by maximal volume decomposition.

It is noted that sub-volumes generated by maximal volume decomposition always volumetrically intersect, which is a useful property for generating multiple feature interpretations. Fig. 11 illustrates the maximal volume decomposition for the delta volume of the machined part in Fig. 10. This delta volume is decomposed into 179 sub-volumes.

3.2 Ordering and generation of tool volumes

One of the important considerations in solid model simplification is the criterion of which features should be removed at a given level of simplification. The criterion varies according to areas of applications, but possible criteria include the feature type and the feature size (volumetric size). With the feature type criterion, a group of features of the same type such as holes, fillets, and slots will be removed upon an interactive request by user. With the feature size criterion, features of smaller size will be removed prior to features of larger size regardless of the feature types.

To create a part by machining, in general, manufacturing engineers first recognize the portions of material to be machined from the final drawing or the final geometry of CAD model. And then they consider various factors such as the number of setups, tools available and type of features to optimize the machining time in determining the sequence of machining operations. A general rule, nevertheless, is that they usually machine the features of a large volume prior to the features of a small volume unless they have certain

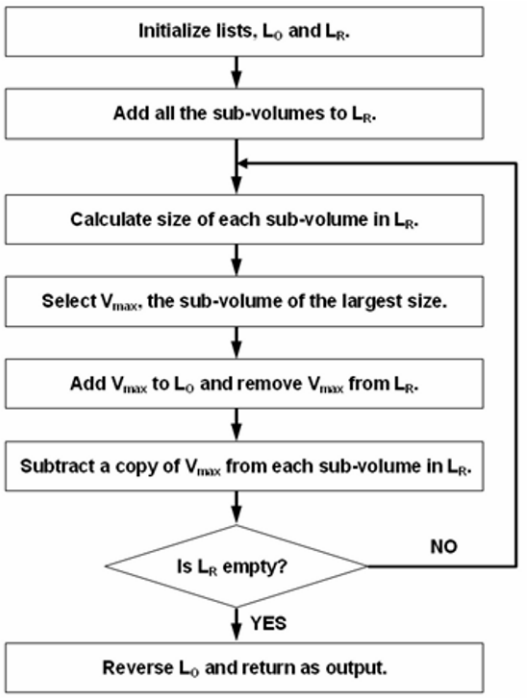


Fig. 12. Ordering and generation of tool volume.

precedence constraints. It is because not only a face of a large feature is often used as a datum of geometric tolerance, but also it is more advantageous to achieve the accuracy of a finished part. In this work, mimicking this machining process planning, the criterion of feature size is used in ordering subtractive features generated by maximal volume decomposition.

As described earlier, a sub-volume generated by maximal volume decomposition intersects with one or more other sub-volumes. That is, a sub-volume added to or removed from the part changes the shapes of other sub-volumes intersecting with it. Thus, it is necessary to generate non-intersecting sub-volumes that actually influence simplifications of the solid model. These non-intersecting sub-volumes are called tool volumes. Tool volumes are generated along with ordering of the sub-volumes, as described in the flow chart in Fig. 12. L_O and L_R in the chart denote an ordered list of sub-volumes and a list of remaining sub-volumes, respectively. Note that the order of tool volumes is reversed in the last process in the flow chart. By doing this, the tool volumes are ordered in ascending order, from tool volume of smaller size to tool volume of larger size. Fig. 13 shows an example of generation of the tool volume of a part.

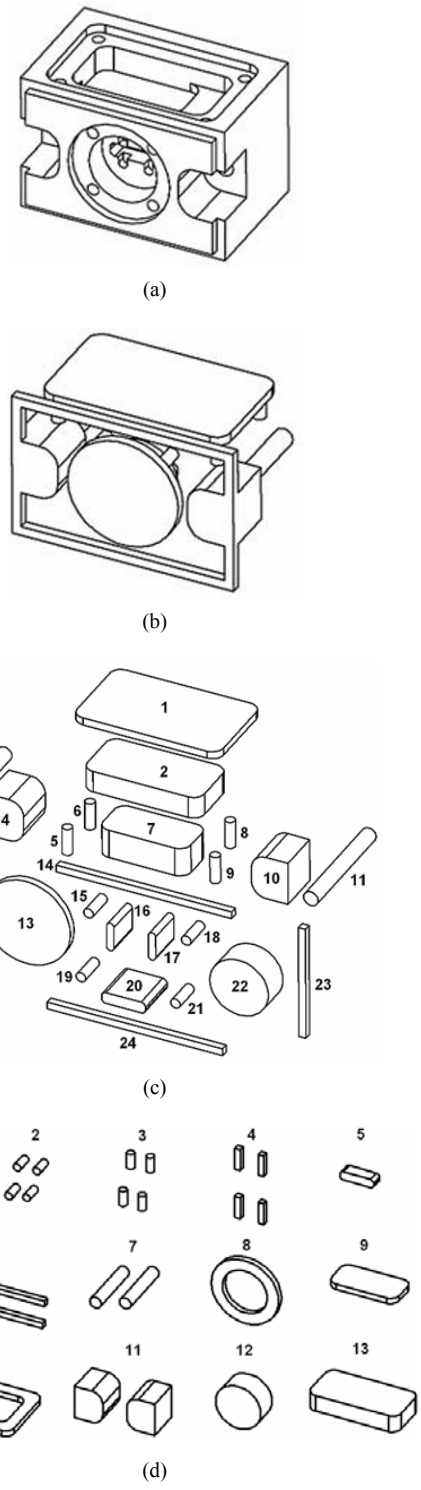


Fig. 13. Ordered tool volumes for a mechanical part; (a) part; (b) delta volume; (c) the ordered set of the sub-volumes of the delta volume; (d) the ordered tool volumes.

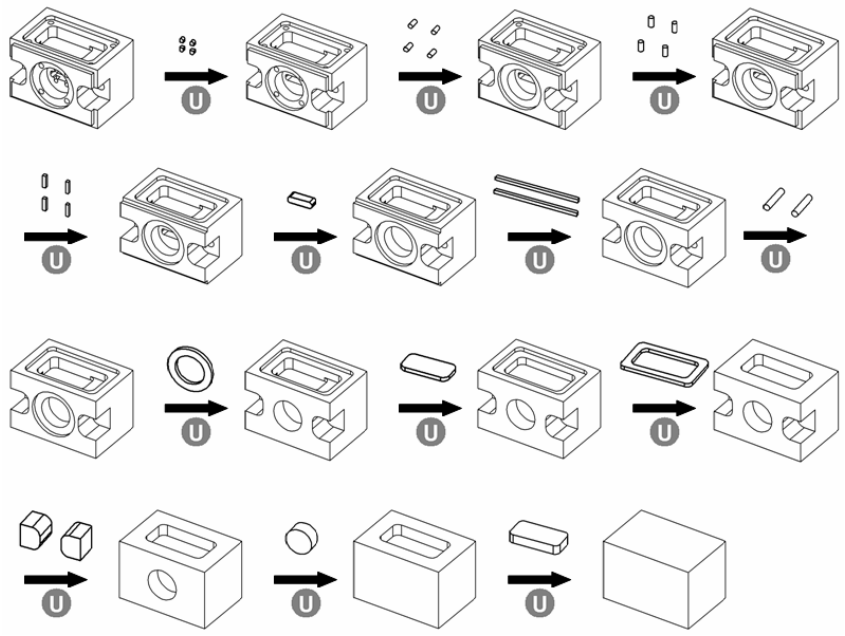


Fig. 14. Simplification of the part in Fig. 13 using the tool volumes.

3.3 Generation of simplified model

Once the tool volumes are generated in ascending order with respect to their size, it is straightforward to create a simplified model at a given level of simplification. The level of simplification is measured by the ratio of the number of tool volumes to be used for simplification to the total number of tool volumes. That is, the level of simplification, L_S, is:

$$L_S(\%) = \frac{\text{Number of tools volumes to be applied}}{\text{Total number of tool volumes}} \times 100$$

A simplified model is generated by uniting the tool volumes with the original solid model up to a given level of simplification. Thus, the 0% simplified model is the original solid model, and the 100% simplified model is the blank workpiece. When the delta volume of solid S is decomposed into N tool volumes denoted by T_i, the simplified model M_{LS}, at the level of simplification L_S, is represented as follows:

$$M_{Ls} = S \cup T_1 \cup T_2 \cup T_3 \ldots \cup T_{\text{int}(Ls \times N/100)} = S \cup \left\{ \sum_{i=0}^{\text{int}(L_S \times N/100)} T_i \right\},$$

where T_0 is a null solid. As an example, the simplified models for the part in Fig. 13 are illustrated in Fig. 14.

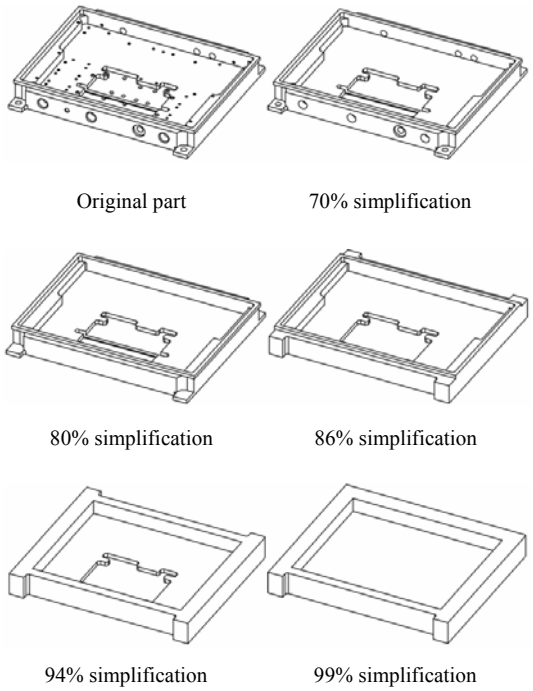


Fig. 15. Example of simplification of machined part.

The method described so far has been implemented as a system using C/C++ on top of ACIS running on a

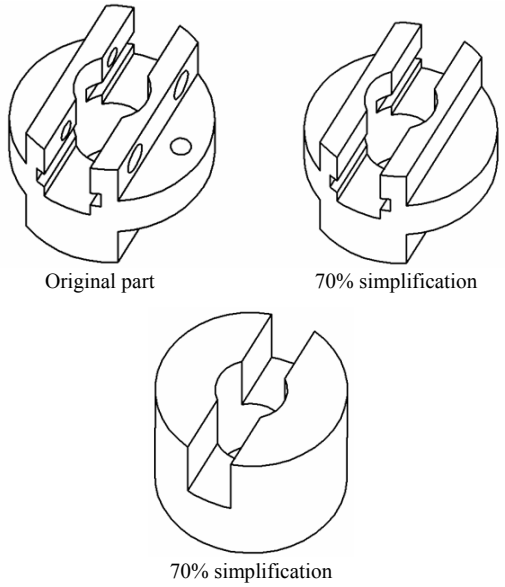


Fig. 16. Example of simplification of machined part.

Windows PC. In this system, a user can interactively specify the level of simplification with a scroll bar. Fig. 15 and Fig. 16 show the examples of the simplification by using the method presented in this paper. The results of these examples attest to the usefulness of the method presented in this paper.

4. Conclusions

This paper has presented a decomposition-based method for simplification of a solid model of machined part. The contributions of this work are summarized as follows:

- This method can generate consistent and acceptable simplification results independent of design features. A consistent result needs to be guaranteed for the same model regardless of designers' modeling preferences and usage of features. Moreover, this method can also be applied to the solid models in neutral formats such as STEP and IGES, since it recognizes features directly from the final geometry of parts.
- This method can recognize intersecting features in the form of volumes and they are readily available for removal of features from the model. Other methods based on graph-matching often fail to recognize intersecting features, and removing faces of features from the model requires additional geometric operations such as tweaking, which often fails due to topological changes in the model.

Due to the nature of the maximal volume decomposition, a solid model that does not have any concave edges cannot be decomposed into sub-volumes. Therefore, if a solid model created by additive features such as a pipe arrangement does not have any concave edges (for example, a case where all the pipe segments are connected by blends), decomposition does not occur. In such cases, preliminary processes such as removal of blending faces may be necessary.

In addition, the method presented in this paper is mainly focused on mechanical parts that will be created by machining. That is, the most simplified result of a solid model will be the solid model of stock material. If a solid model is not appropriate to be represented in terms of machining features, unnatural simplification may happen for the solid model. To address these undesirable cases, one future work is to extend this method so that it can be applied to solid models of non-machined parts.

Acknowledgment

This research was financially supported by Hansung University in the year of 2009.

References

[1] B. Li and J. Liu, Detail feature recognition and decomposition in solid model, *Computer Aided Design*, 34 (2002) 405-414.
[2] K. Shimada, Current trends and issues in automatic mesh generation, *CAD'06 conference*, Phuket Island, Thailand, (2006).
[3] S. Arabshahi and D. C. Barton, Towards integrated design and analysis, *Finite Elements in Analysis and Design*, 9 (1991) 271-293.
[4] M. S. Shephard, E. V. Korngold, R. R. Collar and P. L. Baehmann, A modeling framework for controlling structural idealizations in engineering design, *Computers and structures*, 37 (2) (1990) 181-191.
[5] S. Venkataraman and M. Sohoni, Blend recognition algorithm and applications, *ACM solid modeling conference*, (2001).
[6] S. Koo and K. Lee, Wrap-around operation to make multi-resolution model of part and assembly, *Com-*

puters & graphics, 26 (2002) 687-700.

[7] Y. Lee and K. Lee, Geometric detail suppression by the Fourier transform, *Computer Aided Design*, 30 (9) (1998) 677-693.

[8] S. Kim, K. Lee, T. Hong, M. Kim, M. Jung and Y. Song, An integrated approach to realize multi-resolution of B-rep model, *ACM solid modeling conference*, (2005).

[9] J. Y. Lee, J. Lee, H. Kim and H. Kim, A cellular topology-based approach to generating progressive solid models from features – centric models, *Computer Aided Design*, 36 (3) (2004) 217-229.

[10] S. H. Lee, Feature-based multiresolution modeling of solids, *ACM Transactions on Graphics*, 24 (4) (2005) 1417-1441.

[11] D. Marcheix and G. Pierra, A survey of the persistent naming, *Proceedings of the 7th ACM symposium on Solid modeling and applications*, (2002) 13-22.

[12] R. Bidarra and W. F. Bronsvoort, Persistent naming through persistent entities, *Proceedings of the Geometric Modeling and Processing*, (2002).

[13] H. Sakurai, Volume decomposition and feature recognition: part 1- polyhedral objects, *Computer Aided Design*, 27 (11) (1995) 833-843.

[14] H. Sakurai and P. Dave, Volume decomposition and feature recognition: part II-curved objects, *Computer Aided Design*, 28 (6/7) (1996) 519-537.

[15] Y. Woo and H. Sakurai, Recognition of maximal features by volume decomposition, *Computer Aided Design*, 34 (3) (2002) 195-207.

Yoonhwan Woo is an assistant professor of Mechanical Systems Engineering at Hansung University in Korea. He received his Ph.D. in Mechanical Engineering from Colorado State University in 1999, the M.S. in Mechanical and Aerospace engineering from Illinois Institute of Technology in 1995, and the B.S. in Precision Mechanical Engineering from Hanyang University, Korea, in 1993. He also has been a software engineer of ACIS development team in Spatial Corporation, USA. His research interests include geometric and solid modeling and computer aided process planning.